Advisory For Developing An ACPI-enabled BIOS Under Windows 98

Revision 0.2

September 10, 1998

Intel Corporation

Table of Contents

1. Introduction

Various groups within Intel have been involved in enabling Instantly Available PC capability under Windows 98. Integrating ACPI, Windows 98 and power-management support in various Wake-up devices (all of which are new technologies) has unearthed ambiguities in the specification as well as its interpretation by the various components. This document describes Intel's findings on specific issues in Instantly Available PCs and tips and/or work-arounds that can be implemented in the ACPI-enabled BIOS.

1.1 Overview

The current release of Windows 98 includes support for power management as described in the ACPI and OnNow specifications. These specifications describe how the system can enter a sleeping state to preserve power when the system is idle and then wake from that sleeping state when necessary. See the Reference Documents section for a list of relevant specifications.

Power-managed PCs running Windows 98 can go into any of several different sleep states.
While in this sleep state, in the presence of power-managed peripherals, these systems can be awakened due to external events including button-push, incoming modem calls (Wake-on-Ring) and network-based triggers (Wake-on-Lan, although this capability is not supported in the current version of Windows98).

The environment in which Windows 98 will run is changing as well. New devices will become available that deal with new transport media (xDSL, CableModems). Such devices might be accessed in a variety of different ways (USB, PCI, Ethernet-based). Also, platforms based on newer chipsets will become available that imply new capabilities such as supporting system wake-up by USB-based devices.

A key requirement for power managed PCs supporting Wake-on-Lan and Wake-on-Ring is that these features should work reliably. These PCs, whether clients or servers, should successfully transition between sleeping and waking states. Power managed devices (Wake-on-Lan enabled NICs, power-managed PCI modems) should reliably wake the PC on incoming events while in the sleeping state (wake on LAN via MagicPacket™ frames for Wake-on-LAN, incoming calls for Wake-on-Ring).

This document lists issues that we have found in enabling capabilities in various key areas including:

- Enabling Wake-on-Ring from internal PCI modems
- Enabling Wake-on-Ring from external modems
- Enabling Instantly Available PCs to work with UDMA drives (when DMA is enabled)
- Enabling the graphics subsystem while resuming from S3 sleep state

The rest of this document describes details on these issues and the suggested work-arounds.

1.2 Goals of this advisory

The following are the goals for this specification:

- This advisory is specific to Windows98. The goal is to share information that we have with BIOS developers in order to enable the successful deployment of Instantly Available PCs.

1.3 Non-goals of this advisory

The following are non-goals of this specification:

- This advisory does not cover any findings in WinNT5.0. It is envisioned that a similar document will be made available if the need arises.

1.4 Target Audience

This specification is targeted at BIOS developers and OEMs who are enabling Instantly Available capability in motherboards and PCs.

1.5 Reference Documents

- Advanced Configuration and Power Interface Specification (ACPI), available at http://www.teleport.com/~acpi/
- PCI Spec Rev 2.1 (available at http://www.pcisig.com/specs.html)

2. Details

2.1 Wake-on-ring from an external modem

2.1.1 Symptoms

On an ACPI-enabled system with an external modem attached, when the system goes to sleep in S3-state while a comm application is waiting for an incoming call, the system does not wake-up completely on an incoming call. Specifically, the application (such as HyperTerm) will not get the

ring indication and will wait forever even though the rest of the system is apparently awake. All outstanding TAPI connections need to be closed and reopened for the comm port to be usable.

However, the application works fine when the system goes to S1 sleep-state (where power is not removed to the comm port).

2.1.2 Severity and impact

This has been observed on desk-top as well as lap-top systems.

Severity is HIGH.

2.1.3 Why does it happen?

The UART registers are not being saved and restored at the beginning and the end of the sleep cycle respectively. We have verified that independently saving the registers before entry into S3 sleep state and subsequently restoring the registers before Windows 98 resumes, will cause the application to respond to the incoming call.

2.1.4 What is the work-around in the BIOS?

The UART registers need to be saved by the BIOS before the system goes into S3 state. The BIOS needs to restore the UART registers from the saved state, before handing control to the OS when resuming from S3 state.

2.2 Wake-on-ring from an internal modem

2.2.1 Symptoms

Internal (PCI) modems do not wake-up an ACPI-enabled system on an incoming call when the system is in S3 sleep state (when a comm application, such as HyperTerm is waiting on an incoming call). The system needs to be rebooted before it can be used again.

This problem occurs during S1 sleep states as well, although the failure is not as consistent.

2.2.2 Severity and impact

This has been observed on desk-top systems.

Severity is HIGH.

2.2.3 Why does it happen?

There is a bug in Windows 98 which results in the Configuration-Manager losing track of the reference counts that it uses to arm and disarm wake-up devices.

This problem also causes other wake-up devices (such as Network Interface cards) to not respond to incoming wake-up events.

2.2.4 What is the work-around in the BIOS?

All devices capable of waking up the system need to have a _PRW defined. Specifically, USB device definition (in the ACPI tables) needs to have a _PRW defined for the USB device.

2.3 Wake-up from S3 with UltraDMA drives

2.3.1 Symptom

When DMA is enabled (from the Device Manager U/I) on Windows 98 in a desk-top system (with an UltraDMA drive), the system will hang while waking-up from S3 state. However, the system wakes up fine when DMA is disabled (i.e., defaults to PIO mode in the OS as well as the hard-drives). We have seen this problem with UltraDMA drives from multiple vendors.

However, when the system goes to S1 sleep-state, it wakes up correctly.

2.3.2 Severity and impact

This has been characterized on desk-top systems although the problem is generic and should occur on all systems.

Severity is HIGH.

2.3.3 Why does it happen?

This has been isolated to specific errors in the O.S. code which programs the drive on wakeup.

2.3.4 What is the work-around in the BIOS?

The BIOS needs to support the following feature conditionally (i.e., should be enabled/disabled from the BIOS menu at boot-up time, with the default being `enabled'). If the feature is enabled, the BIOS must do the following. When waking up from S3 sleep-state AND if the drive is an UltraDMA drive, the BIOS needs to test the BUSY bit of the IDE drive until it is cleared by the drive (and RDY bit is set). Only after this has

occurred should the BIOS hand control over to the OS. This can be implemented by reading the Status Register (0x1f7) on the drive, using ATA commands.

The reason that this feature must be made conditional is as follows. When this feature is enabled to work around the problem, the full spin up time of the drive is added to the system resume latency while awaking from S3 sleep state. After this problem is fixed in the OS this additional latency is not desirable and detracts from the system user's experience. Hence it should be made possible to turn off this feature after a newer version of the OS is installed on the system.

2.4 Enabling graphics for wake-up from S3

2.4.1 Symptom

When waking up from S3 sleep-state, graphics is not fully enabled. From the user's perspective, the Windows User interface will not reappear.

However, when the system goes to S1 sleep-state, it wakes up correctly.

2.4.2 Severity and impact

This has been characterized on desk-top systems with the Intel 440BX chip-set (PCI Device ID = 7190h)
with either AGP or PCI graphics cards installed.

Severity is HIGH.

2.4.3 Why does it happen?

This has been isolated to a bug in Windows 98 where the AGP Host Controller register-values are not restored by the AGP Host Controller driver (PCI.MP).

2.4.4 What is the work-around in the BIOS?

The BIOS needs to support the following feature conditionally (i.e., should be enabled/disabled from the BIOS menu at boot-up time, with the default being `enabled'). If the feature is enabled, the BIOS must do the following.

Prior to Suspending to S3 sleep-state, the BIOS needs to save the values of the following registers:

- ATTBASE Device 0, Function 0 PCI Configuration Offset B8-BBh
- APBASE Device 0, Function 0 PCI Configuration Offset 10-13h
- Device 0, Function 0 PCI Configuration Offsets 40-50h
- Device 0, Function 0 PCI Configuration Offsets 52-FFh
- Device 0, Function 0 PCI Configuration Offset 51h

Upon Resuming from S3 sleep-state, the BIOS needs to restore the values of the Device 0, Function 0 in the following order:

1. ATTBASE Device 0, Function 0 PCI Config Offset B8-BBh
2. APBASE Device 0, Function 0 PCI Config Offset 10-13h
3. Device 0, Function 0 PCI Configuration Offsets 40-50h
4. Device 0, Function 0 PCI Configuration Offsets 52-FFh
5. Device 0, Function 0 PCI Configuration Offset 51h

It is important that ATTBASE and APSIZE are restored prior to register 51h. Also 51h should be written to last (after all other registers have been restored).